IPv4 Receiver Module Software Test Bench Instructions

Requirements:

Software:

- Vivado SDK 2016.4
- Terminal program (e.g. Tera-Term)
- Appropriate uart drivers

Hardware:

- Zynq based development board (e.g. MicroZed, ZedBoard)
  - This was implemented for the MicroZed 7010
- JTAG programmer (e.g. Digilent HS2, Digilent HS3, Xilinx HW-USB-II-G)
- Cables necessary for uart communications

Project Files:

- IP_RX_SW_TB.xpr.zip

Inputs:

- System.hdf
  - Software test bench hardware platform specification file
- Design_1_wrapper.bit
  - Software test bench bit stream file
- IP_RX_TESTSUITE
  - Software test bench application project
  - Helloworld.c is main file with all code in it
- IP_RX_TESTSUITE.bsp
  - Software test bench application project board support package

Outputs:

- This test bench produces no archived output, only terminal output
  - Terminal output is performed with xil_printf, set up terminal accordingly to catch xil_printf

Procedure:

Getting Project Files:

1. Download “IP_RX_SW_TB.xpr.zip.001” thru “IP_RX_SW_TB.xpr.zip.008”.
2. Utilizing your choice of compression software (e.g. 7-zip) combine the zip parts

Test Bench files and documentation written by: Justin Samstag

Setup:

1. Place “IP_RX_SW_TB.xpr.zip” in appropriate sandbox area.
2. Create empty folder in sandbox area for SDK workspace.
3. Start Xilinx SDK 2016.4.
4. When prompted to choose a workspace navigate to the empty folder created in step 2.
5. Select the “import project” option from the main splash screen.
6. Select the “Select archive file” radial button and navigate to “IP_RX_SW_TB.xpr.zip”
7. Select to open all three project files found.
    a. Design_1_wrapper_hw_platform_0
    b. IP_RX_TESTSUITE
    c. IP_RX_TESTSUITE_bsp
8. Select Finish.
9. Connect development board to PC with JTAG programmer and uart cable.
10. In Xilinx SDK choose “Xilinx Tools -> Program FPGA”
11. Open terminal program and connect to development board uart.

Test Suite:

1. Right click IP_RX_TESTSUITE in the project explorer pane.
2. Select “Run As -> Launch on Hardware (GDB)”.
3. Look at terminal program to see the results of running the test suite.
    a. See description below for what is happening.

Description:

Hardware:

The hardware in the software test bench consists of a block design implementing several IP Cores. The Zynq PS7 core is implemented with a clock of 100 MHZ. The Zynq PS7 core is connected through an AXI memory interconnect to an AXI Stream FIFO with 8192 entries. The AXI-stream ports of the AXI Stream FIFO are connected to AXI width converters to change the 32 bit wide stream to a 64 bit wide stream. The AXI width converters are connected to two AXI Stream data fifos with 64 bit wide AXI-Stream ports and 8192 entries. The AXI Stream data FIFO cores are connected to the input and output stream of the IPv4 Receiver module Wrapper IP Core.

Utilization of MicroZed 7010:

LUT: 24%

LUTRAM: 20%

FF: 12%

BRAM: 85%

BUFG: 3%

To Target another development board you must open the project in vivado and target the appropriate board and re-generate the bit stream.

Software:

Helloworld.c implements the software test bench. The method RunTestCase() performs all the necessary transactions to write the packet to the hardware, receive the response and check the response for correctness. The return value of the RunTestCase() method implements the XST_SUCCESS and XST_FAILURE constants to determine if the test case response was as expected.

All packet data for the test suite is stored in static C arrays. To run a new test case a C array of u32 must be created to hold the packet data in chunks of 32 bits utilizing the following data format:

[(byte3 byte2 byte1 byte0), (byte7 byte6 byte5 byte4), (byteN byteN-1, byteN-2, byteN-3)]

e.g.:

```
u32 testCase1[] = { 0x1C000045, 0x00400000, 0x90C71101, 0x9E01A8C0, 0xFAFFFFEF, 0x6C0722A1,
                    0x69570800};
```

Array size must be limited for the application to run properly as the heap size is not large enough for the maximum length packet. This boundary case is confirmed to work utilizing the HDL test benches.

The AXI Stream FIFO does not support stream packing so the IPv4 header word with the protocol byte is passed in full to the software test bench. The TKeep signal for the IP Core is connected to the data_out_valid signal from the ip_rx module. The values of data_out_valid are confirmed as correct utilizing the HDL test benches.

The terminal output will show the response data in full for each test case and then at the end of the test suite the result of each test case is summarized for quick reference.